Question 2
Write a report on your understanding of Rendering and Design Patterns. Mention and elaborate where a particular Rendering pattern is applicable and is well suited for which use case.
Rendering Patterns

→

Rendering Patterns

1. **Client-Side Rendering:**
2. **Incremental Static Generation:**
3. **Stepwise Hydration:**
4. **Selective Hydration:**
5. **React Server Components:**
6. **Server-side rendering:**
7. **Static Rendering:**
8. **Server-side streaming rendering:**

Client-Side Rendering:
When a website uses client-side rendering, your browser is responsible for rendering the web page for you. It's fast at first, but can get slower as you progress because it depends on your browser to do so much work.

Incremental Static Generation:
Think of Incremental Static Generation as a fast and reliable website that can show you new content instantly. It prepares pages in advance but updates them dynamically as needed.

Stepwise Hydration:
It's like creating a website step by step. It starts with the basics and then adds more interactive features to make pages load faster and get better over time.

Selective Hydration:
Selective Hydration refers to the skillful use of the web. It saves time and speeds up the process by loading beautiful interactive content only when you need it.

React Server Components:
React server components are like the building blocks of a website. It is designed to make large websites run more efficiently by doing some of the heavy lifting on the server.

Server-side rendering:
Server-side rendering is when the server creates a web page and sends it to your browser. It makes homepages load faster and helps search engines, but it can increase the server's workload.

Static Rendering:
Static rendering is like having a ready-made web page. These are pre-made and can be shipped quickly as their stock is waiting to be discovered.

Server-side streaming rendering:
Server-side streaming rendering will present various parts of the web page to you as they become available. This way your browser can start showing your page immediately and make it faster.

Design Patterns

1. **Compound pattern:**
2. **HOC Pattern (Higher Order Component):**
3. **Hooks Model:**
4. **Container/Presentation Pattern:**
5. **Render Props Pattern:**

1.Compound pattern:
A compound pattern is a design that combines several designs to solve complex problems.
They are dependent on each other with shared state and logic

It leverages the power of various models to solve complex software architecture problems.

2.HOC Pattern (Higher Order Component):
Higher Order Component (HOC) pattern is a widely used design pattern in Reat.js.

This helps us to reuse patterns throughout our applications

HOC is component that receives another component, also It uses logic that are used to pass as a parameters, after applying logic it passes and return the element with additional logic

3.Hooks Model:
The Hooks model, introduced in React 16.8, revolutionizes state management and side effect management in functional components.
Hooks like useState and useEffect allow developers to handle state events and li fecycle events on object, eliminating the need for multiple class objects.

The Hooks pattern increases the readability and ease of using code for developing React applications.

4.Container/Presentation Pattern:
Container/Presentation Pattern is a design pattern that is divided into two main types: container and presentation.

Containers manage state and logic, while presentation elements focus solely on rendering UI elements. This separation increases the clarity and security of the code and makes it easier to test and reuse in front-end development.

5.Render Props Pattern:
Render Props Pattern is a design pattern in React that involves using functions as props for objects.

This function, called render prop, allows the element to share code or state with its children.

Render prop mode facilitates integration and code reuse by customizing prop to specific need, thus promoting flexibility and extending code structure.